IN THE CLAIMS:

1. (Original) An integrated testing method capable of performing a test procedure coucurrently in a multitasking manner on a number of computer components through sorfware simulation, the integrated testing method comprising the steps of:

specifying a total number of simulated operations for testing the components;

specifying a FIFO buffer size for the components;

generating a command sequence including a number of commands based on a first specified random number range, with each command being used to simulate a certain task;

generating a start time of operation based on a second specified random number range;

concurrently activating all the components under test are competing for access to a certain resource, activating an arbiter to perform arbitration for these competing components.

2. (Original) The integrated testing method of claim 1, wherein the resource is a PCI bus.

3. (Original) The integrated testing method of claim 1, wherein the resource is a memory unit.

4. (Original) The integrated testing method of claim 1, wherein the components under test include a DMA component, a SIO component, an ISA component, an AC component, an USB component, an IDE component, an AGP component, a PCI component, and a CPU.

5. (Original) The integrated testing method of claim 1, wherein the arbiter is a South Bridge chipset.

6. (Original) The integrated testing method of claim 1, wherein the arbiter is a North Bridge chipset.

7. (Newly Added) A testing method in a multitasking manner for testing a number of components through software simulation, the testing method comprises:

concurrently activating said components by at least one command to simulate a certain task, when two or more components are competing for access to a resource, activating an arbiter to perform arbitration for these competing components.

8. (Newly Added) The testing method of claim 7, wherein said at least one command is generated from a command sequence, with each command in said sequence being used to simulate a certain task.

9. (Newly Added) The testing method of claim 7, wherein said arbiter is activated to perform arbitration between said competing components and to put said competing components into idle state.

10. (Newly Added) The testing method of claim 7, wherein the resource is a PCI bus.

11. (Newly Added) The testing method of claim 7, wherein the resource is a memory unit.

12. (Newly Added) The testing method of claim 7, wherein the components include a DMA component, a SIO component, an ISA component, an AC component, an USB component, an IDE component, an AGP component, a PCI component, and a CPU.

13. (Newly Added) The testing method of claim 7, wherein the arbiter is a South Bridge chipset.

14. (Newly Added) The testing method of claim 7, wherein the arbiter is a North Bridge chipset.

15. (Newly Added) A testing method in a multitasking manner for testing a number of components through software simulation, the testing method comprises:

A1 concurrently activating said components by at least one command in a random command sequence on a random start time of operation to simulate a certain task, when two or more components are competing for access to a resource, activating an arbiter to perform arbitration for these competing components.

16. (Newly Added) The testing method of claim 15, wherein said at least one command is generated from said random command sequence, with each command in said sequence being used to simulate a certain task.

17. (Newly Added) The testing method of claim 15, wherein said arbiter is activated to perform arbitration between said competing components and to put said competing components into idle state.

18. (Newly Added) The testing method of claim 15, wherein the resource is a PCI bus or a memory unit.

19. (Newly Added) The testing method of claim 15, wherein the components include a DMA component, a SIO component, an ISA component, an AC component, an USB component, an IDE component, an AGP component, a PCI component, and a CPU.

20. (Newly Added) The testing method of claim 15, wherein the arbiter is a South Bridge chipset or a North Bridge chipset.